Lab 1 – C++ Review

# introduction

In this course, students are at the beginning of their computer science careers. In four or so years, though, students will be graduating and looking for jobs on websites such as ZipRecruiter™ and Indeed™. This application implements functionality that one would find on these job posting websites by processing information about programming jobs in the Knoxville area. The data the program will use for testing was taken and simplified and modified for this lab from multiple sources on real job postings in Knoxville, fall 2021.

# Detailed Information

The application asks the user for an input file name. The contents of a sample input file with two jobs is shown below. One job consists of a line with the job description followed by a line with the number of job skills desired, *n*, followed by *n* lines each listing a skill, followed by a salary followed by a company name – all on separate lines. Assume all string data will have less than 256 characters, there will never be more than 10 job skills, and there will not be extra white space in the input file. Also, if there is a job title on one line, the following lines of data for that job are guaranteed to exist. Job titles, skills, and company names may all be comprised of more than one word (use getline).

UI Web Developer

8

Vue, Angular, or React

D3

Javascript

Java

Python

Node.js

Git

Relational database

83200

ITR

Front-End Developer

8

Agile

HTML5

CSS3

Javascript

Angular

NodeJS

AJAX

Git, Bitbucket, or Mercurial

0

CGI Group, Inc.

After getting the file name from the user, the program reads the data from the file into an array of structures using a structure definition such as the one below. The program must be able to process at most data on 10 jobs, but the input file may have more or less entries in it. If the input file has more than 10 jobs in it, the program only reads the first 10. If the input file has fewer than 10 jobs, the program partially fills the array of structures.

const int MAX\_SKILLS = 10;

const int MAX\_STRING\_LENGTH = 256;

struct Job {  
 char jobTitle[MAX\_STRING\_LENGTH];

int numSkills;

char skills[MAX\_SKILLS][ MAX\_STRING\_LENGTH];

double salary;

char company[MAX\_STRING\_LENGTH];

};

After reading the job data into the array of structures, print the following menu for the user:

1 Search for job by title

2 Search for a job by skill

3 Quit

Option>

If the user chooses option 1, the program searches for job by title by asking the user for a job title and printing a neatly formatted table of all jobs with that contain the word in the title (case-insensitive) in alphabetical order by job title as shown below. Words should be left aligned, numbers right aligned.

Job title: developer

Job Title Salary Company

-------------------------------------------------------------------------------

Front-End Developer $ 0 CGI Group, Inc.

Agile

HTML5

CSS3

Javascript

Angular

NodeJS

AJAX

Git, Bitbucket, or Mercurial

UI Web Developer $ 83200 ITR

Vue, Angular, or React

D3

Javascript

Java

Python

Node.js

Git

Relational database

If the user chooses option 2, the program searches for a job by skill by asking the user for the skill (case insensitive) and printing the job title, salary and company of all job postings that require that skill in a neatly formatted table as shown below in alphabetical order by job title. Words should be left aligned, numbers right aligned. Search for any skill that contains the search term.

Job skill: angular

Job Title Salary Company

-------------------------------------------------------------------------------

UI Web Developer $ 83200 ITR

Vue, Angular, or React

D3

Javascript

Java

Python

Node.js

Git

Relational database

A basic outline for your program is below. It is a good, high-level algorithm for the main function where most bullet points are functions themselves.

* Print hello message
* Read the data from the file and place it in the array of structures
* Sort the array of structures by title
* do
  + Print the user menu and get response
  + If response is 1, search by job title and print
  + Otherwise if response is 2, search by skill and print
  + Otherwise if not 1, 2 or 3 print invalid
* while response not 3
* Print goodbye message

Error checking:

* If the input file doesn’t exist, print the error message: Input file \_\_\_\_\_ does not exist. Please contact the administrator. (where \_\_\_ is the filename). End the program with a goodbye message.
* If the input file exists but is empty, print the error message: Input file \_\_\_\_ is empty. Please contact the administrator. (where \_\_\_ is the filename). End the program with a goodbye message.

The easiest way to ask whether a C-string contains another C-string is to use the *strstr* function. This function is just another C-string function such as *strcmp* or *strcpy*. See <https://cplusplus.com/reference/cstring/strstr/?kw=strstr> for more information. If that’s not enough for you to figure out how to use this function, then ask your instructor.

The easiest way to ask whether a C++ string object contains another C++ string object is the string class’ find function. See <https://cplusplus.com/reference/string/string/find/> for more information. If that’s not enough for you to figure out how to use the function, then as your instructor.

Use constants where appropriate. Write and use at least three functions other than main.

See Instructor Best Practices for documentation requirements for comments and style.

# Sample Execution

Welcome to the Jobs.com

Jobs file: Knoxville

1 Search for job by title

2 Search for a job by skill

3 Quit

Option> 1

Job title: developer

Job Title Salary Company

-------------------------------------------------------------------------------

Front-End Developer $ 0 CGI Group, Inc.

Agile

HTML5

CSS3

Javascript

Angular

NodeJS

AJAX

Bitbucket, Git, or Mercurial

UI Web Developer $ 83200 ITR

Vue, Angular, or React

D3

Javascript

Java

Python

Node.js

Git

Relational database

1 Search for job by title

2 Search for a job by skill

3 Quit

Option> 2

Job skill: angular

Job Title Salary Company

-------------------------------------------------------------------------------

UI Web Developer $ 83200 ITR

Vue, Angular, or React

D3

Javascript

Java

Python

Node.js

Git

Relational database

1 Search for job by title

2 Search for a job by skill

3 Quit

Option> 3

Thank you for using Jobs.com.

# Relevance

The code of this lab requires you to use one function, *strstr*, that is very similar to functions you should already be familiar with (e.gs., *strcpy, strcmp*) but that you probably haven’t used before. This is by design and, after reading about it to try to figure it out by yourself, feel free to ask the instructor how to use it.

1. Why is it important to learn how to read through language references such as cplusplus.com?

The code of this lab could be part of some web pages for the recruiting websites such as ZipRecruiter™ and Indeed™.

1. What other data is typically included in a job posting besides what is included in this lab?

# Rubric

For any credit on this lab, the program MUST read data into an array of Job structures from an input file, use this array, and have and use at least three functions other than the main function.

* (10 points) Comments and style
* (4 points) Relevance questions
* Program correctness (86 points)
  + (7 points) Constants
  + (4 points) Non-existent file
  + (4 points) Empty file
  + (15 points) File with fewer than 10 and more than 10 jobs
  + (18 points) Searching by title
  + (18 points) Searching by skill
  + (10 points) Job output sorted by job title
  + (10 points) Miscellaneous

# What to turn in

1. **Turn-in document:** Fill out the turn-in document that is part of this assignment and turn it in via the online course assignment tool.
2. **Code:** Copy your executable a.out file to the instructor’s turn-in directory on the Linux server.